SOME RECENT SIMPLIFICATIONS OF THE THEORY OF FINITE AUTOMATA

RICHARD C. JEFFREY

TECHNICAL REPORT 219

MAY 27, 1959

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
CAMBRIDGE, MASSACHUSETTS

The Research Laboratory of Electronics is an interdepartmental laboratory of the Department of Electrical Engineering and the Department of Physics.

The research reported in this document was made possible in part by support extended the Massachusetts Institute of Technology, Research Laboratory of Electronics, jointly by the U. S. Army (Signal Corps), the U. S. Navy (Office of Naval Research), and the U. S. Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA36-039-sc-78108, Department of the Army Task 3-99-06-108 and Project 3-99-00-100.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 219 May 27, 1959

SOME RECENT SIMPLIFICATIONS OF THE THEORY OF FINITE AUTOMATA

Richard C. Jeffrey

Abstract

Automata are defined as four-termed relations (sets of quadruples). Reduced automata, that is, automata in which the states serve as output symbols, are defined as three-termed relations (sets of triples). A method is given for replacing the nodes and arrows in the graphs of such relations by neurons or by other logical elements in such a way that the resulting net realizes the corresponding automaton. This method is applicable whether or not the relation corresponds to a single-valued function; that is, whether or not the next state of the automaton is uniquely determined by the present state and input symbol. Kleene's theory of the representation of events (sets of input words) by finite automata is presented in a simplified and strengthened form largely due to Medvedev, Nerode, and Rabin and Scott. It is also proved that a set which does not contain the "word" of length zero is representable by a reduced finite automaton if and only if it is representable by a nonreduced finite automaton.

1. INTRODUCTION

In the past three years, work by Medvedev (2), Copi, Elgot, and Wright (3), Nerode (4), and Rabin and Scott (5) has radically simplified and broadened the theory of representation of sets of "words" by finite automata as provided, for example, by Kleene (1). In this report, which is largely a refinement of an attempt made in the fall term of 1958-59 (in course 6.531, M.I.T.) to improve Kleene's exposition (1), we have leaned heavily on the work of references 2, 3, 4, and 5.

Details of certain of the proofs are new, as is the proof in section 8, and, I believe, the technique for realizing automata by nerve nets described in section 3. This has the advantage over Kleene's (1) technique that the structure of the automaton is mirrored in that of the representing net. The method also gives physical significance to the very useful notion introduced by Rabin and Scott (5) of a "non-deterministic" automaton: a nonprobabilistic automaton for which the present input symbol and present state do not uniquely determine the next state. (In effect, Medvedev (2) uses this same notion in proving that all regular sets are representable.)

2. DEFINITIONS

An automaton is a nonempty set of quadruples, finite or infinite. The quadruple whose successive members are x, s, t, and y will be written "$\langle x, s, t, y\rangle$" or, when there is no danger of confusion, "xsty". The terminology for components of a quadruple is: $\langle$input symbol, present state, next state, output symbol$\rangle$. For an automaton A we write:

X_A = A's input alphabet = the set of all first components of quadruples of A.
Y_A = A's output alphabet = the set of all last components of quadruples of A.
S_A = A's states = the set of all second or third components of quadruples of A.
D_A = A's determinants = the set of all pairs that begin quadruples of A.

(We shall sometimes omit the subscript A.)

We shall say that two automata, A and B, are isomorphic if there exist functions f_X, f_S, and f_S with the property that the function f_A maps A onto B in a one-to-one manner, where f_A is defined: $f_A(xsty) = \langle f_X(x), f_S(s), f_S(t), f_Y(y)\rangle$ for all xsty in A.

An automaton A is said to be: unitary, if no two quadruples of A have the same determinant; complete, if D_A is the Cartesian product $X_A \times S_A$; reduced, if for every xsty and x's't'y' in A, y = y' if s = s'. A reduced automaton is one in which the output symbols correspond to sets of states. Accordingly, for some purposes, the output symbols are redundant, and we can often treat reduced automata as sets of triples obtained by reducing xsty to xst.

The graph of an automaton contains one node for each state, and one arrow for each quadruple. The arrow for xsty originates in node s, terminates in node t, and bears the label "x:y". In the graph of a reduced automaton which is being treated as a set of triples, the arrow for xst bears the label "x".

3. REALIZATION OF AUTOMATA BY NETS

The following methods of realization are of interest because they allow the graph of any automaton to be interpreted as an abbreviation for the block diagram of a corresponding net, and so give physical meaning to the constructions, introduced later, in the proofs of theorems.

First, consider a reduced automaton A in the form of a set of triples, with the property that X_A has n members, where $2^{m-1} < n \leq 2^m$. We describe a realization of an automaton B, isomorphic with A, with the property that for each x in X_A, $f_X(x)$ in X_B is an m-tuple of zeros and ones. In terms of nerve nets, each node in the graph of B becomes a neuron (a "node neuron") with threshold 1, and each arrow in the graph of B becomes a neuron (an "arrow neuron") with threshold u + 1, where u is the number of ones in the label of the arrow in question. (See Fig. 1.) The input to the net consists of an ordered m-tuple of "input neurons," each of which has an end bulb impinging on each of the arrow neurons: If the label on the arrow is $\langle x_1, \ldots, x_m \rangle$, then the end bulb from the i^{th} input neuron is inhibitory or excitatory, accordingly as x_i is 0 or 1. (In Fig. 1, to save eyestrain, most of the axons from the input neurons to their end bulbs have not been shown.) It is assumed that all neurons have the same delay, d, and that the period of firing of the input neurons is 2d. The head of each arrow in the graph of

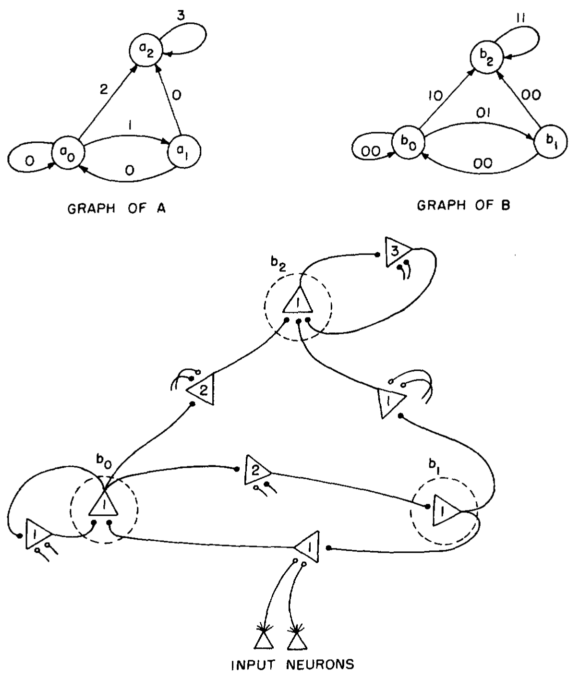

Fig. 1. Illustration of method for interpreting graphs as nerve nets. Nerve-net representation of B. (Input neurons may be state neurons in some other net; in any case, their firing times are those of the state neurons.)

B becomes an excitatory end bulb from the corresponding arrow neuron, impinging on the neuron corresponding to the node on which the arrow terminates. Similarly, the tail of each arrow becomes the body (soma) of the corresponding arrow neuron, on which impinges an excitatory end bulb from the node neuron that corresponds to the node on which the arrow originates.

Logically, there is no need for delay in the arrow neurons. (Alternatively, the node neurons could be delay-free.) Accordingly, the arrow neurons might be replaced by and-gates with inverters (not-elements) at those and-gate inputs that correspond to inhibitory end bulbs from the input neurons, and the node neurons might be replaced by or-gates followed by unit delay elements. In this way, the nerve-net realization can be converted into a realization by a logical net.

If A is a nonreduced, unitary automaton with the property that Y_A has $\underline{o}$ members, we realize an isomorphic automaton B as described above, but with the further stipulation that Y_B shall have $\underline{o}$ members, each of which is an $\underline{o}$-tuple of which one component is 1 and the rest are 0. Physically, the output is represented by the condition (firing or not firing) of an ordered o-tuple of "output neurons" that have a one-to-one correspondence with the output symbols of A: A's output is y if and only if $f_Y(y)$ has a 1 in the position corresponding to y; that is, if and only if the output neuron corresponding to y is firing. Each output neuron has threshold 1; output neuron number i has impinging on it an excitatory end bulb from each arrow neuron which corresponds to an arrow in the graph of B bearing a label of the form $x{:}\langle y_1, \ldots, y_o \rangle$ in which $y_i = 1$ (and all other components are 0). In this mode of realization, the output is obtained 2d units of time after the firing of the input neurons that caused it. By complicating the arrangement and taking advantage of the fact that in a unitary automaton the output is a function of the determinant, the output can be obtained d units of time after the input neuron firing that caused it. (Here, the output neurons are driven by the state neurons and by the input neurons, rather than by the arrow neurons.)

If we now apply the same scheme to a nonreduced, non-unitary automaton, we find that more than one output neuron can be firing at a time: As many as 2^o different firing patterns are possible. For example, for the simple automaton $A = \{0aa0, 0ab1, 1ba2\}$, we construct the isomorphic automaton $B = \{0aa\langle 1, 0, 0\rangle, \; 0ab\langle 0, 1, 0\rangle, \; 1ba\langle 0, 0, 1\rangle\}$. In state a, with input 0, the output of B is 100 and/or 010. We interpret this as a well-defined output, 110, realized physically by the simultaneous firing of the first two output neurons.

This method of realization makes sense out of an automaton's being in two states at once, as well as of the automaton's having two outputs at once. (We identify the state of the net with the over-all firing pattern of its state neurons, and do not require that only one of these be firing at a time. For the corresponding automaton, we identify the over-all states as binary s-tuples, where S_X has s members.)

These considerations justify our definition of an automaton as an arbitrary nonempty set of quadruples, by giving physical meaning to the non-unitary case, and to the incomplete case. (In the incomplete case, it is possible for the state 000 ... 0 to occur; that is, it is possible for all of the state neurons to cease firing.) Finally, arrangements can be made to start a net in any desired state or states by providing an excitatory "starter" end bulb for each state neuron.

4. WORDS; REGULAR SETS

A word on X is defined as a finite sequence of members of X, the length of the word being identified with the length of the sequence. It is convenient to allow the null sequence, I, to count as a word on X (the "identity word," of length 0).

If $u = \langle u_1, \ldots, u_m \rangle = u_1 \ldots u_m$ and $v = v_1 \ldots v_n$ are words on X, we understand the concatenation of u and v, written uv, to be the word $u_1 \ldots u_m v_1 \ldots v_n$. We stipulate that $Iw = wI = w$ for any word w on X.

If U and V are sets of words on X, we define UV as the Cartesian product; that is, UV consists of all words of the form uv, where u is in U and v is in V. $\{I\}$, the set whose only member is I, is then an identity element: $\{I\}W = W\{I\} = W$ for all sets W of words on X. ϕ, the null set of words on X, acts as a zero; that is, we stipulate that $\phi W = W\phi = \phi$ for any set W of words on X.

We shall consistently use u, v, and w to denote words, and U, V, and W to denote sets of words, and shall omit the qualification "in X," whenever it seems obvious. In many contexts, it is convenient to ignore the difference between a word w and the corresponding unit set $\{w\}$. In particular, we define concatenation between a word and a set: $wW = \{w\}W$, and $Ww = W\{w\}$.

Nonnegative powers are defined: $W^0 = \{I\}$, and $W^{i+1} = WW^i$, and similarly for words, $w^0 = I$ and $w^{i+1} = ww^i$.

The closure of a set is defined as

$$W* = \bigcup_{i=0}^{\infty} W^i = \{I\} \cup W \cup WW \cup \ldots$$

where $\cup$ is the set-theoretical union. Similarly, we define $w* = \{w^0, w^1, w^2, \ldots\}$. If we extend the definition of $\cup$ so as to treat w as $\{w\}$ when it is required, that is, if we define $w \cup W = \{w\} \cup W$, $W \cup w = W \cup \{w\}$, and $u \cup v = \{u\} \cup \{v\}$, then the definition becomes $w* = \bigcup_{i=0}^{\infty} w^i$.

Notice that X* is the set of all words on X, and that $XX* = X* - \{I\}$ (which we shall write as $X* - I$) is the set of all words of positive length on X. (A - B is the set-theoretical difference between A and B, consisting of all those members of A that are not members of B.)

We define the regular sets of words on X as comprising all and only the sets that can be obtained from the elements of X by applying the three operations of concatenation, union, and closure any finite number of times. We assume that X is finite or enumerably infinite. Then there will be a nonenumerable infinity of sets of words on X (i.e., of subsets of X*); but there is only an enumerable infinity of regular sets of words on X. Then "almost all" sets of words on X are nonregular.

5. STATE FUNCTIONS

We define the primitive state function σ_A associated with a unitary automaton A by the formula

$$\sigma_A(x, s) = t \longleftrightarrow \bigvee_y xsty \in A \tag{1}$$

(This is read: "$\sigma_A(x, s) = t$ if and only if for some y, xsty is in A.") Thus, if the present state is s and the input symbol is x, the next state (if any) will be $\sigma_A(x, s)$; if A is incomplete, it may be that there is no next state; that is, σ_A is a partial function, undefined for certain arguments. For a reduced automaton A in the form of a set of triples, Eq. 1 becomes simply: $\sigma_A(x, s) = t \longleftrightarrow xst \in A$.

We now wish to extend σ_A to a general state function in such a way that if w is any word on X_A, $\sigma_A(w, s)$ will be the state (if any) reached from s after applying the successive letters of w as inputs to A. We interpret the leftmost letter of w as the first, and the rightmost as the last. If $w = x_1x_2x_3 \ldots x_n$, where all the x_i are in X, then the states reached after applying the successive initial segments x_1, x_1x_2, $x_1x_2x_3$, ... as inputs to A are, respectively, $\sigma_A(x_1, s)$, $\sigma_A(x_2, \sigma_A(x_1, s))$, $\sigma_A(x_3, \sigma_A(x_2, \sigma_A(x_1, s)))$, ... Here, the $i + 1^{st}$ term of the state sequence is $\sigma_A(x_{i+1}, T_i)$, where T_i is the i^{th} term. Then the appropriate generalization of Eq. 1 is given by the recursion

$$\sigma_A(wx, s) = \sigma_A(x, \sigma_A(w, s)) \tag{2}$$

For generality, we define

$$\sigma_a(I, s) = s \tag{3}$$

(The input I is no input at all.) Now Eqs. 1, 2, and 3 define σ_A as generally as we could wish, for unitary A. (If A is incomplete, there will be a state s of A and an infinite set W of words on X with the property that $\sigma_A(w, s)$ is undefined for all w in W. But if A is complete, $\sigma_A(w, s)$ is defined for all w in X_A^* and all s in S_A.

If A is not unitary, Eq. 1 does not define σ_A as a single-valued function. For such cases, it is convenient to have a "total-state" function, Σ_A, of which the values and second arguments are subsets of S_A. But we arrange to ignore the distinction between an element s of S and the set $\{s\}$:

$$\Sigma_A(x, s) = \Sigma_A(x, \{s\}) = \left\{ t\colon \bigvee_y xsty \in A \right\} \tag{4}$$

("$\{t: \ldots\}$" is read: "the set of all t's that is such that ... .") This definition is extended to arbitrary subsets of S_A in the second-argument position by stipulating that Σ_A be "linear" in its second argument when its first argument is a member of X_A:

$$\Sigma_A(x, S' \cup S'') = \Sigma_A(x, S') \cup \Sigma_A(x, S'') \tag{5}$$

for all $x \in X_A$ and $S', S'' \subseteq S_A$. If we then stipulate that

$$\Sigma_A(I, S') = S' \tag{6}$$

for all $S' \subseteq S$, and that

$$\Sigma_A(wx, S') = \Sigma_A(x, \Sigma_A(w, S')) \tag{7}$$

for all $x \in X_A$, $w \in X_A^*$, and $S' \subseteq S_A$, we can prove by induction on the length of w that

$$\Sigma_A(w, S' \cup S'') = \Sigma_A(w, S') \cup \Sigma_A(w, S'')$$

for all $w \in X_A^*$, $S', S'' \subseteq S_A$; that is, that Σ_A is linear in its second argument with any word on X as its first argument.

Σ_A is now completely defined for all automata A. The significance of the "linearity" property is that we can trace the chain of states, starting from state s in an automaton, independently of the fact that the automaton may simultaneously be in some other state, t, which gives rise to its own chain of subsequent states, as in Fig. 2, after the first input symbol. Once the total state ϕ is reached, all subsequent total states will be ϕ,

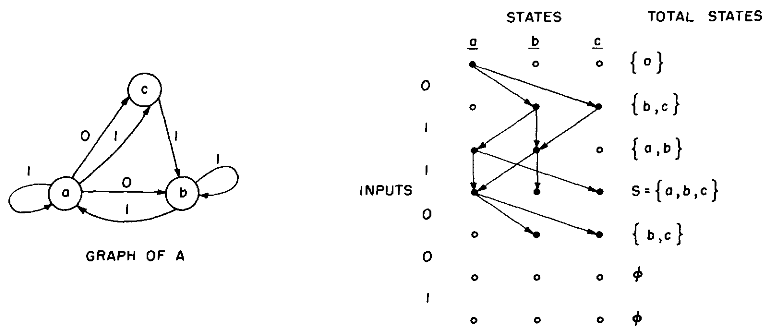

Fig. 2. Illustration of the linearity property: superposition (input word = 011001).

regardless of the input. Notice that instead of identifying the total states with subsets of S we might (equivalently) have identified them with binary sequences; for example, in Fig. 2, $\{a\} = 100$, $\{b, c\} = 011$, and $\phi = 000$.

6. REPRESENTATION OF SETS BY AUTOMATA

The state matrix of an automaton A is defined as the function $M_A(s,t) = \left\{x : \bigvee_y xsty \in A\right\}$. For finite automata, this matrix will be written as a square array of the usual kind. In any case, given an ordering $s_1, s_2, \ldots$ of S, we shall speak of $M_{ij} = (M_A)_{ij} = M_A(s_i, s_j)$

as the "ij component" of M_A; and of the sequence $M_{i1}, M_{i2}, \ldots$ as the "i^{th} row" of M_A, and so forth. M_{ij} is the set of all one-letter words on X that take the automaton from state s_i to state s_j. We now wish to define powers of the matrix M_A so that M_A^n will be the matrix whose ij component is the set of all n-letter words on X that take the automaton from state s_i to state s_j. If M is an a × b matrix, and N is a b × c matrix, and the components of M and N are subsets of X*, then we define the matrix product MN as

$$(MN)_{ij} = \bigcup_{k=1}^{b} M_{ik}N_{kj} \tag{8}$$

where, on the right, juxtaposition indicates concatenation, as defined in section 4. This definition is also applied when $b = \infty$. We define unions of matrices: $(M \cup N)_{ij} = M_i \cup N_i$. The identity matrix is J, where

$$J_{ij} = \begin{cases} I \text{ if } i = j \\ \phi \text{ if } i \neq j \end{cases}$$

We define $M^0 = J$ and $M^{i+1} = M^iM$. Finally,

$$M* = \bigcup_{i=1}^{\infty} M^i = \text{the closure of } M$$

Then the ij component of M_A^* is the set of all words (of whatever length) that take the automaton A from state s_i to state s_j. The intersection of the sets in the j^{th} column of M_A^* is the set of all words that take A into state s_j, regardless of what state A was in to begin with. If a word w appears in none of the sets in the i^{th} row of M_A^*, A is incomplete, and $\Sigma_A(w, s_i) = \phi$. If A is unitary, the sets in the i^{th} row will be disjoint. If A is complete, they will together exhaust X_A^*. The word I belongs to $\left(M_A^*\right)_{ij}$ if and only if i = j.

We say that a set W of words is S-represented (state-represented) by an automaton A, starting in state S_i, if W is the union of certain of the $\left(M_A^*\right)_{ij}$(that is, if W is the "union" of one or more of them).

If A is a unitary automaton, we define its primitive output function, ω_A, as follows (omitting the subscript A):

$$\omega(x, s) = y \leftrightarrow \bigvee_t xsty \in A \tag{9}$$

The general output function is defined by Eq. 9 and Eqs. 10 and 11.

$$\omega(I, s) = I \tag{10}$$

$$\omega(wx, s) = \omega(x, \sigma(w, s)) \tag{11}$$

For any $w \in X*$ we then have $\omega(w, s)$ either a member of Y, or undefined (which will happen, in the unitary case, exactly if $\sigma(w, s)$ is undefined). To get the entire output word associated with $w \in X^*$ and a starting state $s \in S$, we define

$$\Omega(I, s) = I \tag{12}$$

$$\Omega(wx, s) = \Omega(w, s)\,\omega(wx, s) \tag{13}$$

We say that a set W of words is O-represented (output-represented) by an automaton A, starting in state s, if for some $y \in Y$, $W = \{w : \omega(w, s) = y\}$.

7. NERODE'S REPRESENTATION THEOREM

For S-representability, there is a valuable theorem that is due to Nerode (3); the present formulation is due to Rabin and Scott (4).

Given a set $W \subseteq X^*$, we wish to know whether W is finitely S-representable; that is, whether there exists a finite automaton A with starting state s_i that S-represents W. The answer is yes, if and only if the equivalence relation E_W, defined as

$$u \, E_W \, v \leftrightarrow \bigwedge_{w \in X^*} (uw \in W \leftrightarrow vw \in W) \tag{14}$$

partitions X* into a finite number of equivalence classes. ("$u \, E_W \, v$" is read: "u stands in the relation E_W to v"; "$\bigwedge_{w \in X^*}$" is read: "for all w in X*. . . .") We use the notation $[u]_W$, or simply $[u]$, for the equivalence class to which u belongs; that is, $v \in [u] \leftrightarrow u \, E_W \, v$. Two words, u and v stand in the relation E_W if and only if they are "W-indistinguishable" as beginnings of words: To a person or to a machine who/that is blind to all characteristics of words in X* except for the single characteristic of membership in W, the words uw and vw look alike, regardless of what w is, as long as $u \, E_W \, v$.

Another necessary and sufficient condition for finite S-representability of W (it is also due to Nerode, through Rabin and Scott) is that W be the union of some of the equivalence classes of some "right invariant" equivalence relation, R, where the total number of equivalence classes into which R partitions X* is finite. An equivalence relation R is said to be right invariant if

$$uRv \rightarrow uwRvw \tag{15}$$

for all u, v, w in X*. Thus, an equivalence relation R is right-invariant, provided that the operation of "right multiplication" by a word w breaks up no homes: Words that were together in a single equivalence class before the operation find themselves together in a single equivalence class (perhaps different from the one in which they started) after the operation.

Adopting the phrase "R is of finite index" as shorthand for the statement that R (an equivalence relation) partitions X* into a finite number of equivalence classes, we summarize the situation, following Rabin and Scott.

The three following conditions are equivalent:

(i) W is finitely S-representable.

(ii) The relation E_W is of finite index. (16)

(iii) W is the union of some of the equivalence classes of some right-invariant equivalence relation R of finite index.

Proof

(i) $\longrightarrow$ (iii), since the relation

$$uR_i v \longleftrightarrow \left(\bigvee_j u, v \in \left(M_A^*\right)_{ij} \vee \bigwedge_j u, v \notin \left(M_A^*\right)_{ij}\right)$$

is a right-invariant equivalence relation of finite index over X^*. (iii) $\longrightarrow$ (ii), for since W is the union of some of R's equivalence classes, $uwRvw \longrightarrow (uw \in W \longleftrightarrow vw \in W)$ or, by definitions 14 and 15, R is a refinement of E_W, which is therefore of finite index. Finally, (ii) $\longrightarrow$ (i), for the unitary automaton A_W of which the state function is $\sigma(u, [v] = [vu]$ S-realizes W, the initial state being $[I]$.

As a corollary we have the information that the automaton A_W gives the most economical S-realization of W, in the sense that every automaton that S-realizes W has at least as many states as A_W. For the states of A_W are the equivalence classes of E_W, so that it suffices to show that if A S-realizes W, then the relation R_i defined in the proof that (i) $\longrightarrow$ (iii) is a refinement of E_W. Suppose, then, that uR_iv for some $u, v \in X^*$. Since R_i is right-invariant, it follows that uwR_ivw for all $w \in X^*$. And since W is the union of some of R_i's equivalence classes, we have $uw \in W \longleftrightarrow vw \in W$ or, by definition 14, $u E_W v$.

8. REDUCTION TO SETS OF TRIPLES

Using condition 16, we can now prove that

If W is finitely O-representable by unitary automata, it is finitely S-representable. (17)

(The converse is easy to prove if $I \notin W$; but no set containing I is O-representable.) This theorem assures us that in studying finitely representable sets, we lose nothing by confining attention to sets representable by reduced automata: the sets of triples are as productive as the sets of quadruples.

Proof

Suppose that for some finite, unitary A and for some i, $w \in W \longleftrightarrow \omega_A(w, s_i) = y$. Define a relation E:

$$u E v \longleftrightarrow [(u \in W \longleftrightarrow v \in W) \wedge uR_iv]$$

where $\wedge$ means "and." Since E is a refinement of R_i, E, like R_i, is an equivalence relation on X^*. Furthermore, E is right-invariant, for from (i) uR_iv and (ii) $u \in W \leftrightarrow v \in W$ we can derive (iii) uwR_ivw and (iv) $uw \in W \leftrightarrow vw \in W$ for all $w \in X^*$. [(iii) follows from (i) by the right-invariance of R_i. The special case of (iv) in which $w = I$ is simply (ii). This leaves the case in which $w \neq I$, where we must make use of the fact that A O-realizes W. By (i), u and v take A from state s_i to the same state, s_k, unless they both take A to no state at all; that is, unless neither $\sigma_A(u, s_i)$ nor $\sigma_A(v, s_i)$ is defined. Then for any word $w \neq I$, we must have $\omega(uw, s_i) = \omega(vw, s_i) = \omega(w, s_k)$, where the equality is taken to mean also that if any of these is undefined, so are the others. In particular, if any of these are equal to y, they all are, so that if either uw or vw is in W, so is the other.] Since E is a refinement of R_i, W must be the union of some of E's equivalence classes. Finally, E has finite index, for by its definition, it can have, at most, twice the index of R_i. Now by conditions 16, W is S-representable. (In fact, we have shown how to get an S-representation of W out of an O-representation.)

9. KLEENE'S REPRESENTATION THEOREM: W is S-representable by finite, unitary automata if and only if it is regular

The "only if" clause follows from the following lemma:

For a finite, unitary automaton A, each of the $\left(M_A^*\right)_{ij}$ is regular (18)

Proof is by induction on q, the number of quadruples in A. For $q = 1$, $A = \{xs_i s_j y\}$, and $\left(M_A^*\right)_{ij} = \begin{cases} \{x\} & \text{if } i \neq j \\ x^* & \text{if } i = j \end{cases}$. In either case, $\left(M_A^*\right)_{ij}$ is regular. Now suppose that the theorem holds for all A with q quadruples. Let A' have q + 1 quadruples, and let A be obtained from A' by deleting a quadruple $xs_i s_j y$. The set of words that map s_i into s_j in A' without using $xs_i s_j y$ at all is $\left(M_A^*\right)_{ij}$. The set of words that map s_i into s_j by using $xs_i s_j y$ exactly once is $\left(M_A^*\right)_{ia} x \left(M_A^*\right)_{bj}$. The set that uses $xs_i s_j y$ exactly twice is $\left(M_A^*\right)_{ia} x \left(M_A^*\right)_{ba} x \left(M_A^*\right)_{bj}$, and so on. Then the set that uses $xs_i s_j y$ 0 or more times is $\left(M_{A'}^*\right)_{ij} = \left(M_A^*\right)_{ij} \cup \left(M_A^*\right)_{ia} x \left(\left(M_A^*\right)_{ba} x\right)^* \left(M_A^*\right)_{bj}$, which is regular by the hypotheses of the induction.

To prove the "if" clause, we introduce the following lemma:

If W is represented by a finite automaton A, starting in states $s_1, \ldots, s_n$ (i.e., in total state $\{s_1, \ldots, s_n\}$), then W is also S-represented by some complete, unitary, finite automaton B, starting in a single state. (19)

Proof

We present a method of constructing B, given A. By hypothesis, there is a set $K \subseteq S_A$ with the property that a word $w \in X^*$ takes A from total state $\{s_1, \ldots, s_n\}$ into one or more of the states in K if and only if w is in W. Define K' as the set of all total states of A which contain members of K; that is, $T \in K' \leftrightarrow T \cap K \neq \phi$. Then

$w \in W \leftrightarrow \Sigma_A(w, \{s_1, \ldots, s_n\}) \in K'$. We may therefore define B by setting $X_B = X_A$, S_B = the set of all subsets of S_A, and $\sigma_B(w, T) = \Sigma_A(w, T)$. W is the set of all words that take B from state $\{s_1, \ldots, s_n\}$ to a state in K'.

To prove the "if" clause of Kleene's representation theorem, we first note that any automaton $\{xsty\}$, where $t \neq s$, represents $\{x\}$, starting in state s. Then, assuming that sets U and V are S-represented by unitary, finite automata A and B, we show how to construct finite automata that represent $U \cup V$, UV, and V*, respectively. By lemma 19, we shall then have proved by recursion that all regular sets are finitely S-representable.

Since any automaton isomorphic with A will do as well as A for purposes of representing U, there is no loss of generality in assuming that S_A and S_B have no members in common. Similarly, by lemma 19, there is no loss of generality in assuming that A represents U, starting in a single state u, and that B represents V, starting in a single state v. Finally, we use F to denote the set of states of A such that for all $w \in X^*$, $\sigma_A(w, u) \in F$ if and only if $w \in U$, and similarly we assume that $\sigma_B(w, v) \in G$ if and only if $w \in V$.

$U \cup V$ is represented by the automaton $A \cup B$, starting in total state $\{u, v\}$. That is, $w \in U \cup V$ if and only if $\Sigma_{A \cup B}(w, \{u, v\})$ has a member in common with $F \cup G$.

UV is represented by the automaton $A \cup B \cup C$, where C consists of all triples of the form xsv, where $xsf \in A$ for some $f \in F$. Here, $w \in UV$ if and only if $\Sigma_{A \cup B \cup C}(w, u)$ has a member in common with G.

Finally, U* is represented by the automaton $A \cup D$, where D consists of all triples of the form xsu which have the property that $xsf \in A$ for some $f \in F$. Here, $w \in U^*$ if and only if $\Sigma_{A \cup B}(w, u)$ has a member in common with F.

References

1. S. C. Kleene, "Representation of events in nerve nets and finite automata," pp. 3-41 of C. E. Shannon and J. McCarthy, Editors, Automata Studies by W. R. Ashby and others (Princeton University Press, Princeton, N. J., 1956). First published as RAND research memorandum RM-704 (15 December 1951, 101 pp.).

2. I. T. Medvedev, On a class of events representable in a finite automaton, Supplement to the Russian translation of Automata Studies by W. R. Ashby and others, Edited by A. A. Liapunov (Publishing Agency for Foreign Literature, Moscow, 1956); an English translation of this supplement, by J. J. Schorr-Kon, appears as Group Report 34-73, Lincoln Laboratory, M. I. T., 30 June 1958.

3. I. M. Copi, C. C. Elgot, and J. B. Wright, Realization of events by logical nets, J. Association for Computing Machinery 5, 181-196 (1958).

4. A. Nerode, Linear automaton transformations, Proc. Am. Math. Soc. 9, 541-544 (1958).

5. M. O. Rabin and D. Scott, Finite automata and their decision problems, IBM J. Research and Development 3, 114-125 (1959).